MASTER IN COMPUTER SCIENCE PROFESSIONAL FOCUS IN DATA SCIENCE

Explaining latent-based models for link prediction in knowledge graphs

Latour, Guillaume

Award date:
2021

Awarding institution:
University of Namur

UNIVERSITÉ DE NAMUR
Faculté d'informatique
Année académique 2020-2021

Explaining latent-based models for link prediction in knowledge graphs

Guillaume Latour

Maîtres de stage : Luis Galárraga Del Prado & Danaï Symeonidou

Promoteur : _____ (Signature pour approbation du dépôt - REE art. 40)
Benoît Frenay

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques.

Abstract

More and more often ML models are criticised for their lack of interpretability. One must be able to understand the decision process of the model that led to the refusal of its mortgage, the diagnosis of a disease, or any legal advice.
The ability to provide an explanation for a prediction is crucial and has been on the spotlight for a moment now.
Link prediction is an interesting task among the knowledge graph realm due to its various applications, *e.g.* user recommendation, fact checking, *etc.*
As far as we know, the methods providing the best results for link prediction are based on embeddings, and therefore are not intrinsically comprehensible by a human.
This work proposes a post-hoc interpretability procedure based on rule mining that retrieves some insights about the models' motivations for the provided predictions.

Keywords: knowledge graph embeddings, link prediction, explicability, rule mining.

Acknowledgments

First I would like to express my special thanks to Dr. Luis Galárraga Del Prado for his warm welcome, his precious advices and his availability during my stay in Rennes and after my return in Belgium through virtual meetings. My presence in Rennes was enlightened by the LACODAM team for which I keep vivid memories of kindness and benevolence.

I am deeply grateful to Pr. Dr. Ir. Benoît Frénay for offering me the opportunity to work on a such interesting subject in collaboration with the INRIA/IRISA laboratory in Rennes. It was a great experience and I learned a lot about knowledge graph completion and more broadly about the world of research.

Thanks to Dr. Danaï Symenidou for her motivational speeches which helped me get on my feet when I needed to.

Last but not least, I would like to thank my family for their presence and their continuous encouragements. Especially to my girlfriend Lucie Nicolas, who supported me and helped me more than I would like to admit. The education and love I have been receiving helped me far more than the last couple of years.

Contents

1 Introduction

Today we can find almost every desired piece of knowledge on the internet. A quick search allows us to keep being informed on the world news, let us remember that the film where Tom Hanks played a man who has to survive on an island is called "Cast away", let us know that the capital of Ecuador is Quito, and many more other things.

The massive growth of information reachable by everyone on the web has promoted some initiatives to gather knowledge in a structured way. How would we be able to collect, store, enrich and exploit such amount of information?

Knowledge bases (KB) and knowledge graphs (KG) are an answer to that question. They allow the storage of statements of the form $p(s, o)$ such as $capital(Belgium, Brussels)$ where the predicate p can be seen as a directed labelled edge from the subject s to the object o.
The use of KGs is very valuable, both for academic and industrial domains through their diverse applications: recommendation systems, facts checking, question-answering, *etc.*

Even if huge efforts are made for collecting the most complete KGs, it is inevitable that some relations or some entities are missing. This ascertainment leads to an important and active task related to KGs, which consists of bringing out a missing relation between two entities based on the actual knowledge: link prediction. A plethora of methods have been developed to achieve this task, and the approaches based on embeddings appear to provide the best results for now.

The major downside of embedding-based methods is their inability to provide a satisfiable explanation of the decision process that leads to one or another link prediction. This opaqueness prevents specialists from corroborating the decision process of the model, or a researcher from debugging its model while working on it.

In this work, we propose a framework to explain embedding-based models for link prediction. This work also provides and compares explanations based on the locality, either in the vicinity of a statement or in the whole KG.

The rest of this master thesis is structured as follows. In section 2, we introduce an overview of knowledge graphs, logical rules and explanations in ML. Section 3 briefly describes the most popular link prediction techniques developed until this day. Section 4 presents the core of this master thesis and explains the algorithms used to train the surrogate model. Section 5 discusses the experimental metrics and datasets as well as the results of these experiments. Finally, section 6 concludes this master thesis and provides some perspectives.

2 Preliminaries

In this section, we describe knowledge graph, their applications and associated tasks. Then we define interpretability and discuss a few methods generally used to explain a model. Finally we describe Horn rules, which are the goal explanations of our framework.

2.1 Knowledge base & knowledge graph

A **Knowledge Base** (KB) is a collection of structured information in the form of assertions. A **Knowledge Graph** (KG) is a collection of structured information using a graph formalism, particularly apt for binary facts.

These knowledge graphs are used in various domains, either academic or industrial with application such as recommendation systems, fraud detection, proof checker, question answering, *etc.*

A KG can be seen as a collection of **facts** (also called *triples* or *statements*) (s, p, o), also noted $p(s, o)$, respectively composed by a **subject** (or *head*), a **predicate** (or *relation*) and an **object** (or *tail*).

Figure 1 is an illustration of a simple KG. This example is composed of two relations ("marriedTo" and "hasChild") and three entities ("Elvis", "Priscilla" and "Lisa").

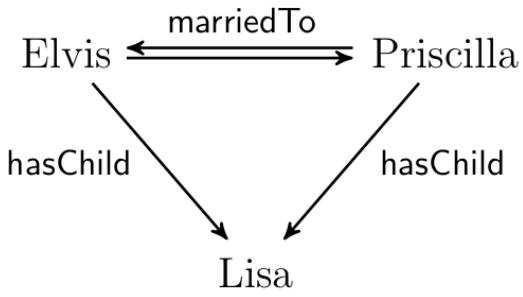


Figure 1: A simple Knowledge Graph. (Source: [1])

Even if KG can be constructed manually, they are generally built using automatic and semi-automatic information extraction methods.

Many current KGs (YAGO, DBpedia, Freebase, WikiData, *etc.*) adopt the *Open World Assumption* (OWA), meaning that missing links cannot be used as false statements. This is in opposition with the *Closed World Assumption* (CWA), for which all missing relation have to be considered as false.
In the middle of these two assumptions, the *Partial Closed World Assumption* (PCWA) mainly treats the knowledge graph like OWA but allows certain parts of the graph to be treated as CWA. This is also known as *Local Closed-World Assumption* (LCWA).

Indeed it is easy to see why storing all false relations is nearly impossible in practice, the relation "marriedTo" alone will create a colossal amount of links between entities since that number will increase quickly with the number of person-type entities.

Since most of the methods for link prediction need some false facts to discriminate between true and false statements, techniques were developed to infer such negative facts from the true facts known by the KG. This is called *Negative Sampling*. The idea is always the same: creating a fact not present in the KG, by fuzzing the head, the relation or the tail of any fact present in the KG. The process chooses randomly which entity will fill the false fact. This randomness is controlled differently by each method. The survey [2] splits the negative sampling methods in three categories: static distribution-based, dynamic distribution-based and custom cluster-based.

2.2 Knowledge graph completion

Even with great effort, KGs will suffer from incompleteness, inconsistencies or incorrectness. This is even harder to avoid as a KG grows in size.

As a way to mitigate these defects, the task of providing missing relations in the KG, and confirming the rightfulness of relations is of prime importance. KG completion may come in different flavours:

- entity prediction, when an element s or o is missing: $(?, p, o)$ or $(s, p, ?)$
- relation prediction, when p is missing: $(s, ?, o)$
- triplet classification, when an algorithm recognises whether a given triple (s, p, o) is correct or not.

The example KG in figure 2 is incomplete. If there is enough data showing that a married couple often has the same children, it is possible that these missing links will be fixed by a KGC task *i.e.* ("Barack" hasChild "Sasha") and ("Barack" hasChild "Malia").

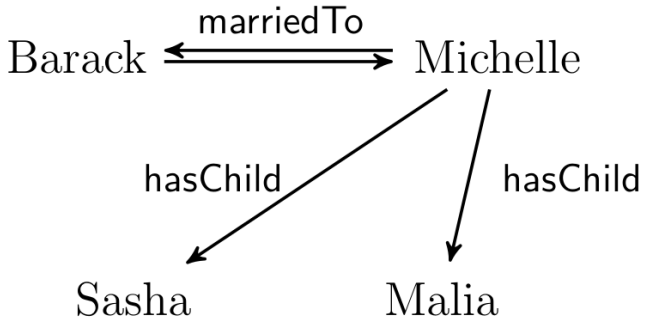


Figure 2: A simple but incomplete KG. (Source: [1])

2.3 Interpretability & explicability

There is no formal definition of interpretability but the concept is defined in [3] as follows: "Interpretability is the degree to which a human can understand the cause of a decision."

There is no absolute threshold value from which a model is interpretable, but the easier it is for a human to understand the decision made by a model, the more this model is interpretable.
In other words, the interpretability of a model can be seen as the degree to which this model can be understood by a human without any help. It is often said that an interpretable model provides its own explanation. It is also referred to as *intrinsic interpretability*.

If a model is not understandable out of the blue but some insights can be gathered through tools and techniques, it is said to be explicable. The explicability can then be seen as the degree to which explanation can be provided for a model's prediction. This is also referred to as post-hoc interpretability.

In the following, we detail some of the existing techniques [4] allowing that kind of post-hoc interpretability.

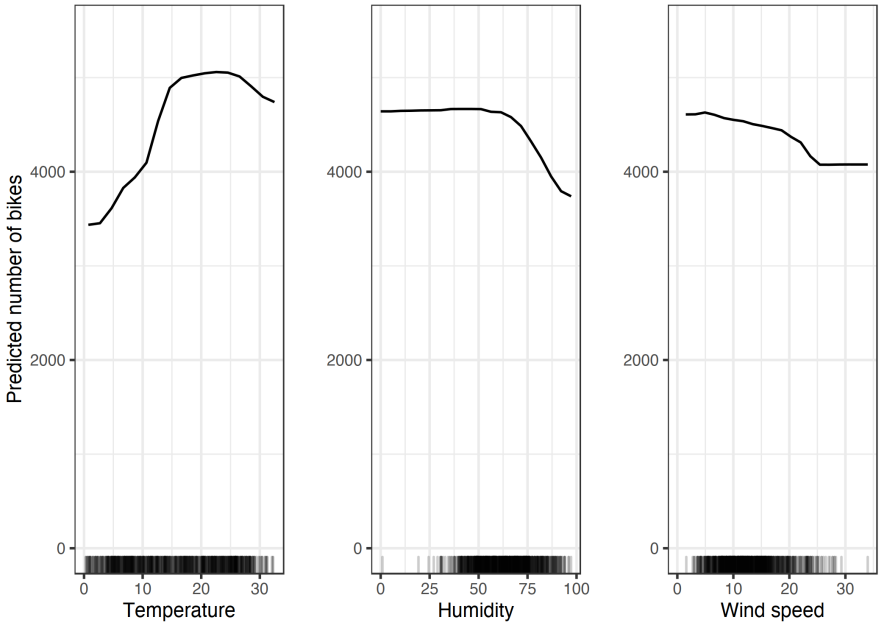
Partial Dependence Plots (PDP) shows the importance a few features have on the predicted target value of the model. The idea is just to print out the relation on a graph between the observed feature and the predicted target value. An example of this plot can be seen in the figure 3a.

Accumulated Local Effects plot (ALE) is the next step in that direction, and achieves the same goal: describing how a feature affects the target value. The ALE plot is better than the PDP because it is informative even if the features are correlated, which can lead to inconsistency between predictions and real values in PDP. Here's an example of ALE plot at figure 3b.

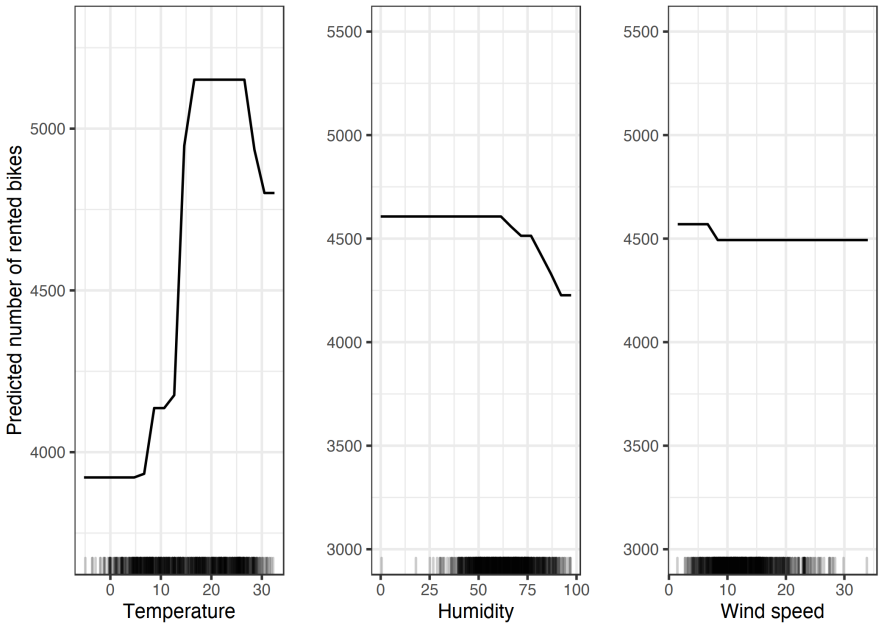
Global Surrogate consists of training a new, interpretable model to approximate the predictions of the black-box model and uses the interpretable predictions to make assertions about the black-box model. The goal of a surrogate model is to have achieve high *fidelity* while being understandable by a human. The fidelity is the accuracy of the surrogate model when the ground truth is given by the black-box model.

Local Surrogate operates under the same principle than the global surrogate model, but here the idea is to train the interpretable model around a single prediction of the black-box model. LIME (Local Surrogate Model-agnostic Explanations)[5] proposes an implementation for these local surrogate models, and outputs a linear model. By restricting the locality around one instance, LIME pledges to achieve *local fidelity*. While it is nearly impossible for an explanation to be faithful of the whole model, LIME focuses on being faithful on the vicinity of a considered instance.

Scoped Rules (Anchors) also have the core intent of providing explanations only in the vicinity of a specific instance. Instead of producing a linear model, the output of this method is a single rule composed by a conjunction of IF-THEN propositions using the features' values for comparison or threshold. This rule is associated with metrics like precision and coverage. This method uses reinforcement learning to explore and evaluate rules around the target instance. The authors [5, 6] provide a visualisation (figure 4) of how differently the two methods (LIME and anchors) conduct around a target instance when trying to extract results.



(a)



(b)

Figure 3: (a) PDP showing the influence of temperature, humidity and wind speed on the predicted number of rented bikes. (b) ALE plot showing the same graphs, but the impact of correlated features are reduced, making it more trustworthy. (Source: [4])

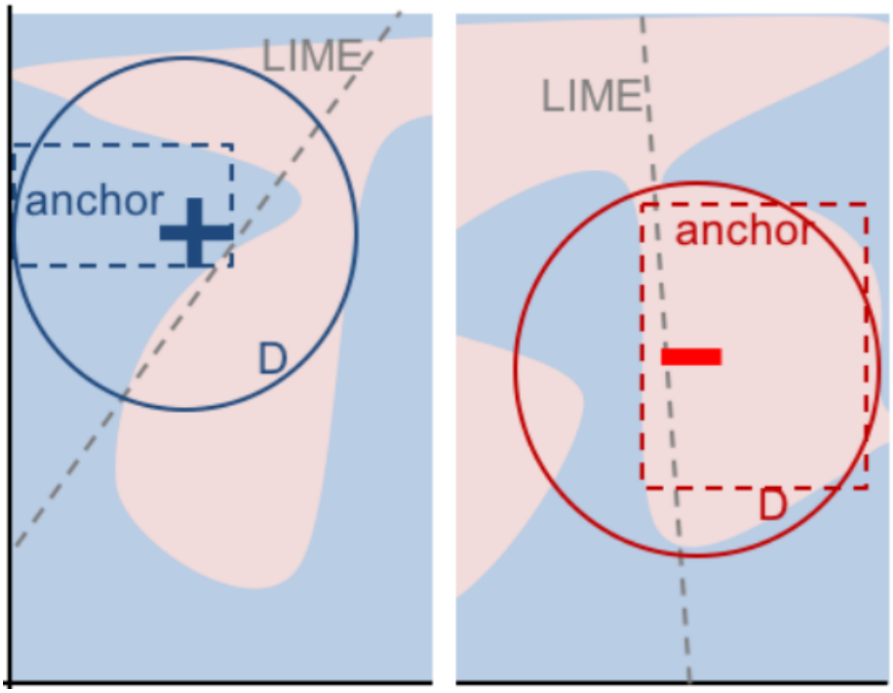


Figure 4: Toy visualisation of the action of anchor and LIME on a binary classifier [6]

Shapley Values comes from cooperative game theory. It considers every feature as a "player", the target value as the "payout" and has the objective of correctly distributing the payout among the player relative to their responsibility in that payout. The result is the importance of each feature for a prediction.

In this work we will be creating a local surrogate model, considering the embedding based model for link prediction as our black-box model and creating contexts of different size to see the impact on the fidelity of the model and the quality of the explanation.

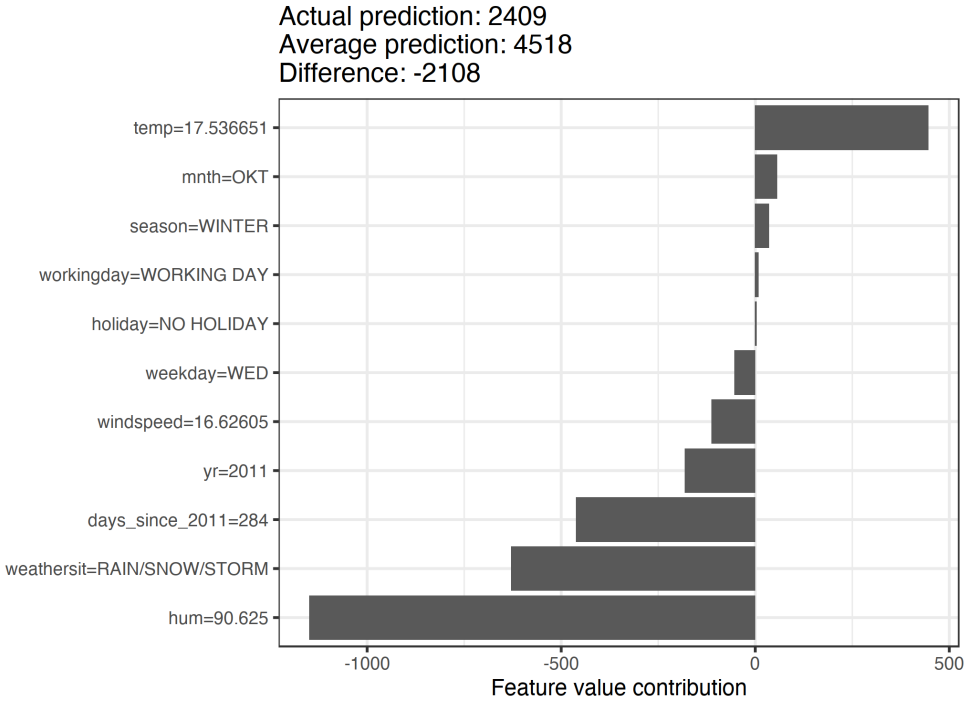


Figure 5: Shapley value for day 285 in the bike rental dataset. The prediction of rented bike this day is 2108 less than the average, and is strongly affected by the humidity and the weather. (Source: [4])

2.4 Horn rules

While it is interesting to see the importance of each features for a prediction as an explanation for this prediction, it is often better to have a human readable explanation for it. In this path, the anchor method is very interesting because even someone who has received no particular training could understand the meaning of the explanation.

(Horn) rules meet this requirement. They are formed with a conjunction of relations and end with an implication. Also, their use are suitable for KGs because a link between two edge can be considered as a binary value, which can be grouped with logical AND. And the presence of a link is understandable without training, so it can be used as explanation.

Atoms

An **atom** is a triple in which the subject and the object can be variables. A different terminology is used depending on the number of variables used in the

atom. If both the subject and object are variables, the atom is **unbounded** *e.g.* $capital(x, y)$. If one of them is a constant, the atom is **bounded** *e.g.* $capital(x, Brussels)$. If there is no variable in the atom, it is **grounded**. A grounded atom is another name for a fact *e.g.* $capital(Belgium, Brussels)$.

Horn rules

A **Horn rule** is an implication that can be written like this: $B \implies H$ where B is called the **body**, which is a conjunction of atoms, and where H is the **head** atom. The following rule can be used as an example: $marriedTo(x, y) \wedge hasChild(x, z) \implies hasChild(y, z)$.
A Horn rule is called **safe** if the variables in the head atoms do appear in the body. The rule $married(x, y) \wedge livesIn(x, z) \implies livesIn(y, z)$ is safe because the two variables of the head (y and z) do appear in the body. $married(x, y) \wedge livesIn(x, z) \implies livesIn(u, z)$ is not safe because one of the variables of the head (u) does not appear in the body.

3 State of the art: Link prediction

The goal of the link prediction task is to infer missing relationships between entities, which can be formulated as predicting the head or the tail of a statement given the other entity and the relation.

In the early days of link prediction, the way to go was to learn inference rules from a sequence of triples [7].
However, this method does not scale well since the number of unique sequences of triples increases with the number of relations and the large sizes of current KGs prevent the use of these methods.

More recently, another family of approaches allows a better generalisation by operating on embedding representations [8]. During training, models learn a scoring function that optimises the score of a target entity for a given triple. In the evaluation, a statement with a missing entity is mapped into the latent space and the model outputs a prediction vector for the missing entity.

A taxonomy of the state-of-the art method can be seen in figure 6.
The Wang et al. survey[9] provides a comprehensive and exhaustive list of the KGE models for link prediction.

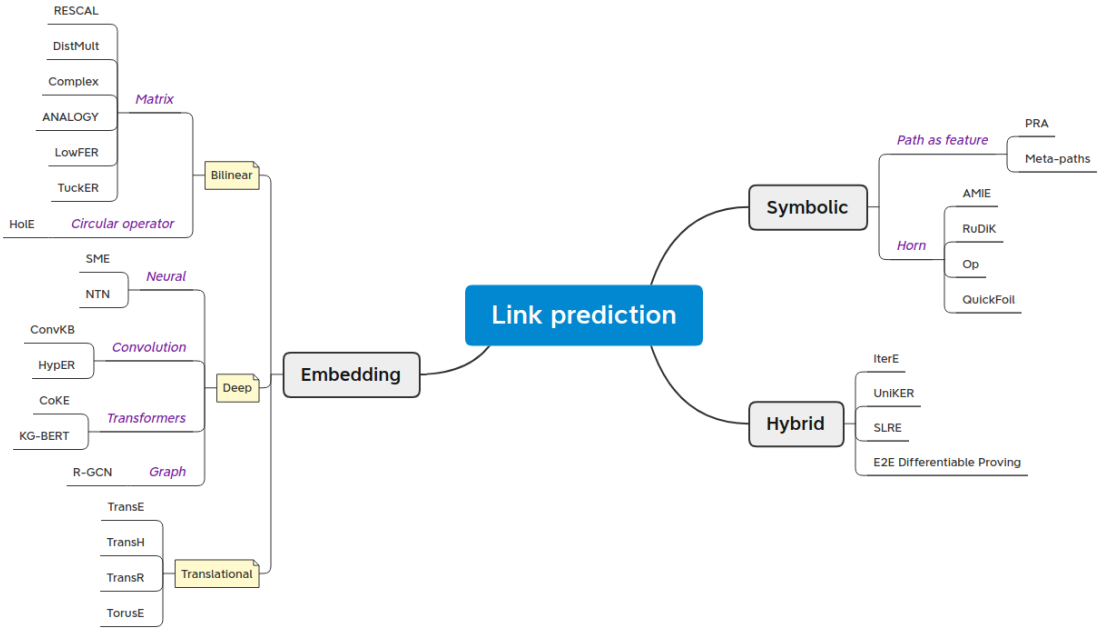


Figure 6: Link predictor taxonomy

3.1 Embedding methods

The embedding models use latent features to predict facts in KGs. The idea is to represent entities *via* embeddings, and predicates as relations between those embeddings.
Embedding-based methods transform a KG into a low-dimensional vector space while preserving its underlying semantics.
The models provide a score function $f(s,p,o)$ which for a given triple $< s,p,o >$ reflects the model's confidence in the truthfulness of the triple. The scoring function and the embeddings are designed so that true triples get a higher score than false triples. Based on this, potential candidates for a given query $< s,p,? >$ can be ranked.

RESCAL[10] is a factorisation-based bilinear model. It represents entities as vectors $a_i \in \mathbb{R}_n$, relations as matrices $R_k \in \mathbb{R}^{n*n}$ and has a score function $f(s,p,o) = a_s^T R_p a_o$.

DisMult[11] has the same working principle as RESCAL but uses a diagonal matrix for its relations R_k. This decreases the number of parameters in the model.

Complex[12] is using a complex diagonal matrix for its relations R_k. The score of $p(s,o)$ is given by the real part of $a_s^T R_p a_o$. It allows a better handling of asymmetric relations.

ANALOGY[13] uses structural composition of a known KG to induce, by analogy, missing links in another, unknown KG. This method optimises the latent representations with respect to the analogical properties of the embedded entities and relations.

LowFER[14] proposes a factorised bilinear pooling model, commonly used in multi-modal learning, for better fusion of entities and relations, leading to an efficient and constraint-free model.

TuckER[15] is a linear model based on the Tucker decomposition[1] of the binary tensor representation of knowledge graph triples.

[1] The Tucker decomposition[16] splits a tensor into a set of matrices and a small core tensor.

HolE[17] (Holographic Embeddings) represents entities as vectors $a_i \in \mathbb{R}_n$, relations as vectors $r_k \in \mathbb{R}_n$ and has a score function $f(s,p,o) = r_p^T(a_s * a_o)$, where $*$ refers to the circular correlation between a_s and a_o.

SME[18] (Semantic Matching Energy function) uses a neural network to determine the embeddings. The training process captures the implicit structure of the knowledge graph.
Another particularity of this model is that the relations are modelled like the entities. Therefore entities may be used as predicates, as in natural language.

NTN[19] (Neural Tensor Networks) represents entities as an average of their constituting word vectors. This allows entities sharing words to be spotted as similar. This model is improved by an initialisation of these word vectors learned from large text corpora.

ConvKB[20] employs a convolutional neural network. ConvKB can capture global relationships and transitional characteristics between entities and relations in knowledge bases. In ConvKB, each triple is represented as a 3-column matrix where each column vector represents a triple element. This 3-column matrix is then fed to a convolution layer where multiple filters are operated on the matrix to generate different feature maps. These feature maps are then concatenated into a single feature vector representing the input triple. The feature vector is multiplied with a weight vector via a dot product to return a score. This score is then used to predict whether the triple is valid or not.

HypeER[21] proposes a hypernetwork[2] architecture that generates simplified relation-specific convolutional filters that improve performances greatly; and can be framed as tensor factorisation and thus set within a well established family of factorisation models for link prediction.

CoKE[22] (Contextualised Knowledge graph Embedding) innovates by taking into account contextual information on the entity and relation embeddings. "Unlike previous methods that assign a single static representation to each entity/relation learned from the whole KG, CoKE models that representation as a function of each individual graph context, *i.e.* an edge or a path."[22]

[2]Approach of using one network (the hypernetwork), to generate the weights for another network.

KG-BERT[23] (Knowledge Graph Bidirectional Encoder Representations from Transformer) uses pre-trained language models for knowledge graph completion and it treats triples in knowledge graphs as textual sequences.

R-GCNs[24] (Relational Graph Convolutional Networks) are a demonstration of the usage of the GCN for the link prediction and triple-classification tasks. It allows to deal with the highly multi-relational data characteristic of realistic knowledge bases. It also assert that other bilinear models are to be improved with the help of an encoder model to accumulate evidence over multiple inference steps in the relational graph.

Translational methods use translation-based model, which represents entities as vectors $a_i \in \mathbb{R}_n$, relations as vectors $r_k \in \mathbb{R}_n$. TransE[25] has a score function $f(s,p,o) = \|a_s + r_p - a_o\|_2^2$. This method can only handle 1-to-1 predicates. TransR[26] and TransH[27] allow to handle 1-to-N, N-to-1 and N-to-N predicates with an increase of the number of parameters as a trade-off.

TorusE[28] was developed to tackle the regularisation issue that appears on TransE. TransE forces entity embeddings to be on a sphere in the embedding vector space while TorusE allows the usage of a Lie group[3].

3.2 Symbolic methods

This family of methods uses two techniques to provide link prediction: either via mining Horn rules; or via the exploitation of the path leading from an entity to another. These kind of prediction are considered intrinsically explainable because what lead to the prediction is understandable by a human.
" Rules are useful in a wide range of applications: knowledge reasoning and expansion [12, 46], knowledge base construction [9], question answering [14], knowledge cleaning [19, 44], knowledge base maintenance [48], Markov logic learning [27], etc " (from [29])

AMIE [30] is a rule mining framework that can find rules on millions of facts in a few minutes without the need for parameter tuning or expert input. It is explicitly tailored to support the OWA scenario since it has a method to simulate negative examples without making a closed world assumption.

[3]A Lie group is a topological space that has the following properties: it is a group and a differentiable manifold

RuDiK[31] is a mining framework that has the ability to discover positive and negative rules. A negative rule could be: if a is Belgian, he does not have a president. This is really helpful to spot the errors in an existing KG.

OP[29] (Ontological Pathfinding) is a mining framework that has a unique partitioning system for the KG that divides the mining task into small and independent sub-tasks, allowing it to run the same algorithm in parallel.

QuickFoil[32] is a method that succeed the scaling of Inductive Logic Programming with a new scoring function and pruning strategy.

PRA[33] (Path Ranking Algorithm) finds paths that often connect entities that are instances of the edge type being predicted. PRA then uses those path types as features in a logistic regression model to infer missing edges in the graph.

Meta-Path[34] exploits the semantic richness of the current knowledge graphs to create path structure between certain types of entities.

3.3 Hybrid

Symbolic reasoning is accurate and interpretable, but it suffers from scale issues.

Embedding-based reasoning is more scalable and efficient as the reasoning is conducted via computation between embeddings, but the sparse entities leads to poor representations because it relies heavily on data richness. The second downfall of embedding-based methods is their lack of interpretability. The Hybrid-based reasoning tends to complement each other's difficulties with their advantages.

IterE[35] learns both rules and embeddings. Rules are learned from embeddings with proper pruning strategy, and embeddings are learned from existing and new triples. The new ones being inferred by rules.

UniKER[36] combines KGE and logical rules for better KG inference in an iterative manner. The authors argue that it is an error to make only a one-time injection of logic rules to KG embeddings because it fails to capture the mutual interaction between KGE and logical rules. They also state that, for

scaling purposes, the other methods use sampling strategies that select only a portion of the rules, which causes loss of information. UniKER pledge to solve these problems.

SLRE[37] (Soft Logical Regularity) allows the use of soft rules in common effort with embeddings for the link-prediction task. Unlike a hard rule (*e.g.* the capital of a country is inside the country), a soft rule may be broken occasionally (*e.g.* the nationality of a person is often the country where he/she was born) This method proposes a highly scalable and effective method for preserving soft logical regularities by imposing soft rule constraints on the relation latent representations.

E2E[38] (End-to-End differentiable proving) replaces symbolic unification with a differentiable computation on vector representations of symbols using a radial basis function kernel, thereby combining symbolic reasoning with learning sub-symbolic vector representations.

4 Algorithms

The main objective is to train a surrogate model that mimics the predictions made by an embedding-based model, considered here as a black-box. Our strategy is to train a linear regression model with rules as features. So we need to extract rules from a KG, and this KG needs to reflect, in some way, the black-box model. The idea is to create contexts labelled by the black-box model, so the truth nature of every facts in the context comes from black-box prediction. Then we merge this labelled context with the KG used to train the black-box model. So our rule miner (AMIE[30]) is influenced by the black-box model and has enough data to work on.

The training of our surrogate model is based on a context, which is local or global and provide different insights. The **local contexts** (algorithm 1) can be viewed as the neighbourhood of a triple and are created around one fact by taking the other existing triples that only differ by their subject or object. The **global contexts** (algorithm 2) gather every triples having their predicate in common.

Both local and global contexts need to be given some counter-examples. Those negative facts are found with Bernoulli negative sampling [27].

Algorithm 1 Local contexts generator

procedure LOCALCONTEXTS($\mathcal{K}_{test}$)
  for each $p(s,o) \in \mathcal{K}_{test}$ **do**
    $C^+ = \{p(s,o)\} \cup \{p(s',o) \in \mathcal{K}_{test}, s' \neq s\} \cup \{p(s,o') \in \mathcal{K}_{test}, o' \neq o\}$
    C^- is populated with Bernoulli negative sampling of C^+
    $C = C^+ \cup C^-$
    yield C

Algorithm 2 Global contexts generator

procedure GLOBALCONTEXTS($\mathcal{K}_{test}$)
  for each $p \in$ RELATIONS($\mathcal{K}_{test}$) **do** ▷ RELATIONS(KG) returns the set of predicates of a KG
    $C^+ = \{p(s,o) \in \mathcal{K}_{test}\}$
    C^- is populated with Bernoulli negative sampling of C^+
    $C = C^+ \cup C^-$
    yield C

Algorithm 3 shows how we train the surrogate model with the following variables. The flowchart describing that algorithm can be found in appendix A.

- A link predictor $\hat{f} : \mathbb{R}^k \to \mathbb{R}$, our experiments focus on the TransE model, but any link predictor can be used.
- A KG $\mathcal{K}_{train} = \mathcal{K}^+_{train} \cup \mathcal{K}^-_{train}$. $\mathcal{K}_{train}$ is the training dataset of the black-box model. $\mathcal{K}^+$ is a set of true facts, $\mathcal{K}^-$ is a set of false facts.
- A context C that can be local or global.
- A rule miner $\mathcal{R}$, our experiments used AMIE but another symbolic method providing rules could have been used.

The first step of our training (lines 2 to 5) is to use the black-box model to label each and every triples of a given context. All the following steps will now consider those label as the truth, since it is given by the model we are trying to approximate.

Then (line 6) we mine the KG, with a filter on the produced rules: the predicate of the context have to appear in the head of the rule. This restriction is a parameter of our rule miner (AMIE) and its purposes are to improve performance and to mine pertinent rules in regard to the context. The KG used for the rule mining step is the union of the black-box's training's dataset and the context that is labelled by this black-box. Every rules mined here has a confidence score associated with.

At that point (lines 7 to 10), we want to create a matrix with the mined rules as features, the facts as instances and the labels as targets. To fill the rules weight for every fact, we check if the rule correctly predict a statement and provide the confidence as weight. If the rule is used in the prediction of another statement, the score is the opposite of the confidence. If the rule has no relation with the statement, a weight of zero is given.
Given A, the considered fact; i the index of the considered rule; $x_A[i]$ the weight of that rule in the prediction for that fact; and R the mined rules.

$$x_A[i] = \begin{cases} conf(R_i) & R_i \wedge (C_{label} \cup \mathcal{K}_{train}) \models A \\ -conf(R_i) & R_i \wedge (C_{label} \cup \mathcal{K}_{train}) \models A' \text{ with } A \neq A' \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

And finally (line 11) we train a surrogate model (linear regression) from this matrix and the weights of the rules impacting the decision will be used as explanation.

We have implemented the contexts generators and the training process in python with the help of the library "torchKGE"[4]. An implementation of our algorithms is publicly available on the INRIA gitlab[5].

Algorithm 3 Train a surrogate model

1: **procedure** TRAINING($\hat{f}, \mathcal{K}_{train}, C, \mathcal{R}$)
2: $C_{label} = \emptyset$
3: **for each** $p(s,o) \in C$ **do**
4: $l \leftarrow f(s,p,o)$
5: $C_{label} = C_{label} \cup \{(s,p,o,l)\}$
6: $R = \mathcal{R}(\mathcal{K}_{train} \cup C_{label})$ ▷ $\mathcal{K}_{train}$ is labelled
7: create an empty matrix M with $|R|$ columns and $|C_{label}|$ rows
8: **for each** $(s,p,o,l) \in C_{label}$ **do**
9: **for each** $r \in R$ **do**
10: $M_r^c =$ weight of r for (s,p,o,l)
11: train model m with R as features, weighted by M and l as target.

[4] `https://torchkge.readthedocs.io/en/latest/`
[5] `https://gitlab.inria.fr/glatour/geebis`

5 Experiments

5.1 Datasets

When it comes to processing knowledge graphs, multiple state-of-the-art datasets can be used to be able to evaluate the performance of a knowledge base link predictor. Those datasets are already split in training, validation and testing set.

fb15k-237
Freebase is a project aiming to gather knowledge across the internet and store it into a knowledge base (graph). This project lived between 2007 and 2015, leaving the place for its successor, Wikidata.
The fb15k dataset was introduced in [25] as a subset of Freebase, with nearly 15k entities. This dataset was found to suffer from overfitting since a lot of test triples could be obtained simply by inverting train triples. This was spotted and fixed by [39], introducing the fb15k-237 dataset.

wn18rr
Wordnet[40] is a massive lexical database in English developed by linguists of the university of Princeton.
wn18 is a subset of Wordnet, containing the 18 most represented relations. In an effort to avoid inverse relations that would arm the results of any link prediction process, wn18rr simply removed the redundant facts from wn18.

Dataset	Predicates	Entities	Facts		
			Training	Validation	Testing
wn18rr	11	40 943	86 835	3 034	3 134
fb15k-237	237	15 541	272 115	17 535	20 466

Table 1: Experimental Datasets

5.2 Evaluation

It is important to understand the different metrics used to gain insights on the model. For our experiments we decided to gather metrics for the *link prediction* and *triplet classification* tasks. The two tasks have specific metrics associated with them, described below.

It seemed important to distinguish the accuracy of the black-box model and the accuracy of the surrogate model. For the replication of the predictions of the black-box, we introduce the term **fidelity**. The fidelity is the accuracy of the surrogate model with the ground truth being the predictions made by the black-box model. The term "accuracy" here has to be understood in its broad sense: every metric could be susceptible to have its equivalent fidelity metric to see to which extend the surrogate can mimic the black-box on a specific task.

The **coverage** is the ratio of facts for which an explanation is provided. It provides confidence in the explanation as a complement for the other metrics. Indeed, even if a given metric is good, a poor coverage will indicate that the prediction has not to be trusted blindly.

The *link prediction* task consist on finding the best answer to a bounded atom. This task can be evaluated with the following metrics: Mean Reciprocal Rank (MRR) and hit@k.

The **rank** of a prediction is calculated like this: for one unbounded atom, the model provides multiple answers sorted from the most to the least probable, the rank of the prediction is the position of the correct answer in this list. *e.g.* with the unbounded atom $capital(Belgium, x)$, if the model answers are "Bern", "Brussels" and "Paris", then this prediction has a rank of 2. The **reciprocal rank** is the inverse of the rank, so if the rank is 2, the reciprocal rank is $\frac{1}{2} = 0.5$. The **mean reciprocal rank** is the average of the reciprocal ranks.

The **hit@k** is the count of good predictions in the top k list of a model's answers, relative to the number of atoms that are seeking prediction.

Triplet classification is a simple classification task. Given triples, the model labels them as correct or incorrect ones. The **accuracy** is the measure of the correctness of the model. It is calculated as the ratio of the correct answers' count by the number of answers given.

5.3 Results

Table 2 shows the overall metrics for our surrogate model approximating the TransE model with the datasets wn18rr and fb15k-237. The accuracy of TransE is pretty low, for any of the two datasets. These results can still be interesting by drilling down to an accuracy by predicates metric. The overall fidelity is

impressively high for the fb15k-237 dataset and quite mediocre for wn18rr. The MRR is correct for wn18rr and worse for fb15k-237.

Dataset	Fidelity	Accuracy(black-box)	MRR	support
wn18rr	0.487	0.536	0.105	41
fb15k-237	0.941	0.506	0.079	8755

Table 2: Dataset comparison for TransE

The table 3 displays metrics about the predictions of our surrogate model emulating TransE on the wn18rr dataset. The *hypernym*[6] relation have better results for both the tasks of triplet classification and link prediction than the other relation *instance_hypernym*. The MMR score of the *hypernym* prediction is the best we have, so we can assume that this model is particularly good at finding the missing entity of an atom.

Predicate	Fidelity	Accuracy (black-box)	MRR	support
hypernym	0.642	0.571	0.151	14
instance_hypernym	0.478	0.608	0.047	23

Table 3: Metrics of the surrogate model mimicking TransE on wn18rr

Table 4 contains the same type of information than the previous table, for the dataset fb15k-237. The fidelity is significantly higher than wn18rr, which indicates that the classification task is easier for this dataset. The MRR in overall is weaker except for the two last entries which have excellent score. The accuracy of TransE is similar to what we have seen before.
Lets analyse the top rules impacting the "film produced by" predicate predictions.

- (a /film/film/produced_by "Adam Sandler")
  $\implies$ (a /film/film/produced_by "Jack Giarraputo")
- ("Rob Schneider" /film/actor/film./film/performance/film a)
  $\implies$ (a /film/film/produced_by "Jack Giarraputo")

Those rules tell us that the black-box model learnt that Jack Giarraputo always works with the same people.

[6]A word is an **hypernym** of another (denoted **hyponym**) if the semantic field of the hyponym is included in the hypernym. *e.g.* "cat", "lynx", "panther", "lion" and "tiger" are all hyponyms of "feline", their hypernym; which is itself the hyponym of "animal", its hypernym.

Predicate	Fidelity	Accuracy	MRR	Support
/people/person/profession	0.975	0.510	0.002	787
/award_nominee/award_nomination	0.998	0.488	0.125	641
/film/film/genre	0.993	0.460	0.066	434
/people/person/nationality	0.942	0.531	0.001	297
/film/film/language	0.931	0.481	0.003	189
/people/lived_in/location	0.994	0.53	1	183
/film/film/produced_by	0.916	0.611	0.285	36

Table 4: Metrics of the surrogate model mimicking TransE on fb15k-237

Here is another rule, for the relation /people/lived_in/location :
(a /has_celebrity_friend b)
$\wedge$ (b /film/actor/film./film/performance/film "The Soloist")
$\Longrightarrow$ (a /people/lived_in/location "Los Angeles")

This rule basically tells that if a has a friend who preformed in the film "The Soloist", a lives in Los Angeles.

The fact that these rules are mined in a global context is surprising because they look highly specific. Maybe there is a too significant amount of films produced by Jack Giarraputo or by the crew of "The Soloist" in the dataset. Or maybe it is a TransE problem that our surrogate model brought to the surface, and this problem would not have been noticed in any other way.
Indeed if the accuracy of the link predictor is low, but the fidelity is high, the rules could be used to debug why the black-box model makes bad decisions.

Next we compare rules obtained from local and global contexts around the predicate "/people/person/profession". Table 5 shows the metrics of the local context around the shown facts while the metrics about the global context for this predicate can be found in table 4.
Here is one rule mined within the global context.

- (a /music/genre/artists "Maroon 5")
  $\wedge$ (a $\neg$ /music/instrument/instrumentalists b)
  $\Longrightarrow$ (b /people/person/profession "Singer/Songwriter")

Which can be understood like this: an artist being in the same musical genre as Maroon 5, and who does not practice an instrument is a singer/songwriter. It makes sense that an artist working in the music industry who does not play

Context neighbour	Fidelity	Accuracy	MRR	Support
“James Taylor” profession “Actor”	1	0.550	0.017	158
“Keith David” profession “Actor”	1	0.518	0.015	158
“Matthew Weiner” profession “Actor”	0.993	0.462	0.016	158
“Robbie Robertson” profession “Actor”	0.987	0.525	0.009	158
“Marilyn Manson” profession “Actor”	0.987	0.512	0.01	158
“Alec Berg” profession “Actor”	0.981	0.462	0.01	158
“Mike Figgis” profession “Film director”	0.98	0.58	0.25	50
“Frank Capra” profession “Film director”	1	0.510	0.204	49

Table 5: Metrics of the surrogate model mimicking TransE on fb15k-237, locally around instances containing the predicate *profession*.

an instrument must be known for his/her voice or songwriting skill.

The following rules are mined within local context, respectively from (“James Taylor” profession “Actor”), (“Keith David” profession “Actor”) and (“Mike Figgis” profession “Film director“).

- (“John Cleese” /actor/performance/film a)
  $\wedge$ (a /film/film/written_by b)
  $\implies$ (b /people/person/profession “Actor”)
- (“Nicole Richie” /has_celebrity_friend b)
  $\wedge$ (a /has_celebrity_friend b)
  $\implies$ (a /people/person/profession “Actor”)
- (a /director/film b)
  $\wedge$ ($b \neg$ /film_release_region “Thuringia”[7])
  $\implies$ (a /people/person/profession “Film director”)

Which can be understood by, respectively:

- if b writes a movie in which John Cleese plays, b is an actor,
- Nicole Richie friends' friends are actors,
- if a directed a film that was not released in Thuringia, a is a film director.

[7] Thuringia is a state of Germany

Apart from the third rule, these are quite specific and linked to the local context in which they are mined. For example it would be a wise guess to say that not all of Nicole Richie's friends are actors, but in the given context it seemed like it was pertinent. The third one is an over-complication of "if a directed a film, a is a film director", and since not many people direct films without being film directors it is a pertinent rule, but not restricted to the local scope of the context.

Some rules can be mirrors of each other and artificially bring erroneous metrics to show up. For instance the following rules were mined in the local context of ("Keith David" profession "Actor"):

- ("Nicole Richie" /has_celebrity_friend b)
  $\wedge$ (a /has_celebrity_friend b)
  $\implies$ (a /people/person/profession "Actor")
- ("Nicole Richie" /has_celebrity_friend b)
  $\wedge$ (b /has_celebrity_friend a)
  $\implies$ (a /people/person/profession "Actor")
- (b /canoodled a)
  $\wedge$ (b /romantic_relationship "Nicole Richie")
  $\implies$ (a /people/person/profession "Actor")
- (b /canoodled a)
  $\wedge$ ("Nicole Richie" /romantic_relationship b)
  $\implies$ (a /people/person/profession "Actor")

Here we have two mirror relations around the predicates *has celebrity friend* and *romantic relationship*. Actually finding those rules help us understand that there is redundancy in the data. The avoidance of the mining of those kind of rules could be perspectives for future works.

6 Conclusion

In this work we have presented the importance of the KG and their link prediction tasks. After going through embedding-based models for link prediction and realised the importance of providing an explanation we introduced our framework as a proposition to palliate this problem. The framework allows any embedding-based model to be approximated by a surrogate model providing explanations. Those explanations are also influenced by the scope in which they have been computed.

Several sources of improvements are available for future works. Two tasks have been discussed in this master thesis and a third one could be the interesting. After triplet classification and entity prediction, the task of relation prediction has yet to be implemented.

The surrogate model is trained by a logistic regression, but any other model able to use its weighted features as explanations could be used here. A comparison between various model could be interesting.

The context creation, especially for the local context, could be reviewed. Multiple algorithm for local context creation could be proposed and compared to each other.

References

[1] Jonathan Lajus, "Fast, Exact, and Exhaustive Rule Mining in Large Knowledge Bases," Ph.D. dissertation.

[2] J. Qian, G. Li, K. Atkinson, and Y. Yue, "Understanding negative sampling in knowledge graph embedding," International Journal of Artificial Intelligence & Applications, vol. 12, pp. 71–81, 01 2021.

[3] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," 2018.

[4] C. Molnar, Interpretable Machine Learning, 2019, https://christophm.github.io/interpretable-ml-book/.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," arXiv:1602.04938 [cs, stat], Aug. 2016, arXiv: 1602.04938. [Online]. Available: http://arxiv.org/abs/1602.04938

[6] ——, "Anchors: High Precision Model-Agnostic Explanations," p. 9.

[7] S. Schoenmackers, J. Davis, O. Etzioni, and D. Weld, "Learning first-order horn clauses from web text," 10 2010, pp. 1088–1098.

[8] Y. Shen, P.-S. Huang, M.-W. Chang, and J. Gao, "Link prediction using embedded knowledge graphs," 2018.

[9] M. Wang, L. Qiu, and X. Wang, "A survey on knowledge graph embeddings for link prediction," Symmetry, vol. 13, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2073-8994/13/3/485

[10] M. Nickel, V. Tresp, and H.-P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data," Proceedings of the 28th International Conference on Machine Learning (ICML), p. 8, 2011.

[11] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," arXiv:1412.6575 [cs], Aug. 2015, arXiv: 1412.6575. [Online]. Available: http://arxiv.org/abs/1412.6575

[12] T. T. et al., "Complex Embeddings for Simple Link Prediction," arXiv:1606.06357 [cs, stat], Jun. 2016, arXiv: 1606.06357. [Online]. Available: http://arxiv.org/abs/1606.06357

[13] H. Liu, Y. Wu, and Y. Yang, “Analogical inference for multi-relational embeddings,” 2017.

[14] S. Amin, S. Varanasi, K. A. Dunfield, and G. Neumann, “Lowfer: Low-rank bilinear pooling for link prediction,” 2020.

[15] I. Balazevic, C. Allen, and T. Hospedales, “Tucker: Tensor factorization for knowledge graph completion,” Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019. [Online]. Available: http://dx.doi.org/10.18653/v1/D19-1522

[16] L. Tucker, “Some mathematical notes on three-mode factor analysis,” Psychometrika, vol. 31, no. 3, pp. 279–311, 1966. [Online]. Available: https://EconPapers.repec.org/RePEc:spr:psycho:v:31:y:1966:i:3:p:279-311

[17] M. Nickel, L. Rosasco, and T. Poggio, “Holographic Embeddings of Knowledge Graphs,” arXiv:1510.04935 [cs, stat], Dec. 2015, arXiv: 1510.04935. [Online]. Available: http://arxiv.org/abs/1510.04935

[18] X. Glorot, A. Bordes, J. Weston, and Y. Bengio, “A semantic matching energy function for learning with multi-relational data,” 2013.

[19] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, “Reasoning with neural tensor networks for knowledge base completion,” in Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS’13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 926–934.

[20] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, “A novel embedding model for knowledge base completion based on convolutional neural network,” Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018. [Online]. Available: http://dx.doi.org/10.18653/v1/N18-2053

[21] I. Balažević, C. Allen, and T. M. Hospedales, “Hypernetwork knowledge graph embeddings,” Lecture Notes in Computer Science, p. 553–565, 2019. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-30493-5_52

[22] Q. Wang, P. Huang, H. Wang, S. Dai, W. Jiang, J. Liu, Y. Lyu, Y. Zhu, and H. Wu, "Coke: Contextualized knowledge graph embedding," 2020.

[23] L. Yao, C. Mao, and Y. Luo, "Kg-bert: Bert for knowledge graph completion," 2019.

[24] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," 2017.

[25] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," p. 9, 2013.

[26] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, ser. AAAI'15. AAAI Press, 2015, p. 2181–2187.

[27] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1, Jun. 2014. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/8870

[28] T. Ebisu and R. Ichise, "Toruse: Knowledge graph embedding on a lie group," 2017.

[29] Y. Chen, S. Goldberg, D. Z. Wang, and S. S. Johri, "Ontological pathfinding," in Proceedings of the 2016 International Conference on Management of Data, ser. SIGMOD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 835–846. [Online]. Available: https://doi.org/10.1145/2882903.2882954

[30] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "AMIE: association rule mining under incomplete evidence in ontological knowledge bases," in Proceedings of the 22nd international conference on World Wide Web - WWW '13. Rio de Janeiro, Brazil: ACM Press, 2013, pp. 413–422. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2488388.2488425

[31] S. Ortona, V. V. Meduri, and P. Papotti, "RuDiK: rule discovery in knowledge bases," Proceedings of the VLDB Endowment, vol. 11, no. 12, pp. 1946–1949, Aug. 2018. [Online]. Available: https://dl.acm.org/doi/10.14778/3229863.3236231

[32] Q. Zeng, J. M. Patel, and D. Page, "Quickfoil: Scalable inductive logic programming," Proc. VLDB Endow., vol. 8, no. 3, p. 197–208, Nov. 2014. [Online]. Available: https://doi.org/10.14778/2735508.2735510

[33] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," Machine Learning, vol. 81, pp. 53–67, 2010.

[34] X. Cao, Y. Zheng, C. Shi, J. Li, and B. Wu, "Meta-path-based link prediction in schema-rich heterogeneous information network," International Journal of Data Science and Analytics, vol. 3, no. 4, pp. 285–296, Jun 2017. [Online]. Available: https://doi.org/10.1007/s41060-017-0046-1

[35] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, and H. Chen, "Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning," arXiv:1903.08948 [cs], Mar. 2019, arXiv: 1903.08948. [Online]. Available: http://arxiv.org/abs/1903.08948

[36] K. Cheng, Z. Yang, M. Zhang, and Y. Sun, "UniKER: A Unified Framework for Combining Embedding and Horn Rules for Knowledge Graph Inference," p. 7.

[37] S. Guo, L. Li, Z. Hui, L. Meng, B. Ma, W. Liu, L. Wang, H. Zhai, and H. Zhang, Knowledge Graph Embedding Preserving Soft Logical Regularity. New York, NY, USA: Association for Computing Machinery, 2020, p. 425–434. [Online]. Available: https://doi.org/10.1145/3340531.3412055

[38] T. Rocktäschel and S. Riedel, "End-to-End Differentiable Proving," arXiv:1705.11040 [cs], Dec. 2017, arXiv: 1705.11040. [Online]. Available: http://arxiv.org/abs/1705.11040

[39] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1499–1509. [Online]. Available: https://aclanthology.org/D15-1174

[40] C. Fellbaum, "Wordnet: An electronic lexical database," 1998.

A Surrogate's training's flowchart

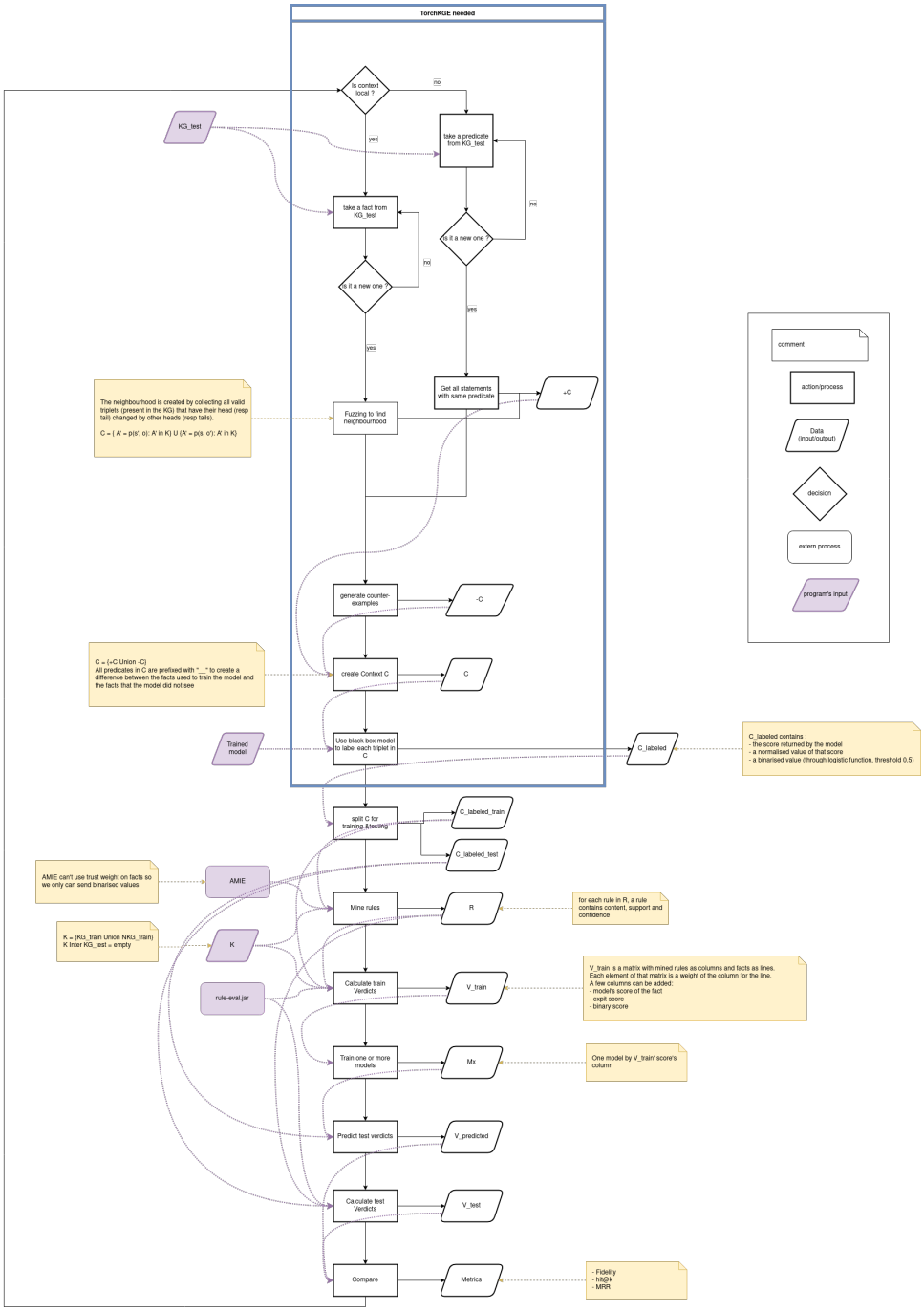


Figure 7: Algorithm used to train proxy model

B Rules mined in local context

For (“James Taylor” /people/person/profession “Actor”)

- (“John Cleese” /actor/performance/film a)
  $\wedge$ (a /film/film/written_by b)
  $\Longrightarrow$ (b /people/person/profession “Actor”)
- (“Michael Palin” /nominated_for a)
  $\wedge$ (a /film/film/written_by b)
  $\Longrightarrow$ (b /people/person/profession “Actor”)
- (a /award_nominee b)
  $\wedge$ (“Monty Python and the Holy Grail” /film/film/written_by b)
  $\Longrightarrow$ (a /people/person/profession “Actor”)

For (“Keith David” /people/person/profession “Actor”)

- (“Nicole Richie” /has_celebrity_friend b)
  $\wedge$ (a /has_celebrity_friend b)
  $\Longrightarrow$ (a /people/person/profession “Actor”)
- (“Nicole Richie” /has_celebrity_friend b)
  $\wedge$ (b /has_celebrity_friend a)
  $\Longrightarrow$ (a /people/person/profession “Actor”)
- (b /canoodled a)
  $\wedge$ (b /romantic_relationship “Nicole Richie”)
  $\Longrightarrow$ (a /people/person/profession “Actor”)
- (b /canoodled a)
  $\wedge$ (“Nicole Richie” /romantic_relationship b)
  $\Longrightarrow$ (a /people/person/profession “Actor”)
- (b /has_celebrity_friend “Lindsay Lohan”)
  $\wedge$ (a /has_celebrity_friend b)
  $\Longrightarrow$ (a /people/person/profession “Actor”)

For (“Mike Figgis” /people/person/profession “film director”)

- (a /film/director/film b)
  $\wedge$ (b /film/film/produced_by “Danny DeVito”)
  $\Longrightarrow$ (a /people/person/profession “film director”)

- (b /award_winner “Steven Soderbergh”)
  $\wedge$ (a /film/director/film b)
  $\Longrightarrow$ (a /people/person/profession “film director”)

- (a /film/director/film b)
  $\wedge$ (b /film/film/cinematography “Steven Soderbergh”)
  $\Longrightarrow$ (a /people/person/profession “film director”)

- (a /film/director/film b)
  $\wedge$ (b neg_/film_release_region “Thuringia”)
  $\Longrightarrow$ (a /people/person/profession “film director”)

- (a /film/director/film b)
  $\wedge$ (b /film/film/edited_by “Steven Soderbergh”)
  $\Longrightarrow$ (a /people/person/profession “film director”)